In the Claims:

1. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree, which Oommen-Rueda tree has a root, branches and leaves, comprising the steps of:

(a) receiving a first character of plaintext;

(b) traversing the Oommen-Rueda tree between the root of the Oommen-Rueda tree, and a leaf of the Oommen-Rueda tree corresponding to the first character of plaintext, and recording an Assignment Value of at least one branch of the Oommen-Rueda tree so traversed;

(c) calculating a character of ciphertext related to the Assignment Value of the at least one branch of the Oommen-Rueda tree so traversed;

(d) calculating an Assignment Value for at least one other branch of the Oommen-Rueda tree related to a distribution of the ciphertext previously calculated;

(e) receiving a next character of plaintext;

(f) traversing the Oommen-Rueda tree between the root of the Oommen-Rueda tree and a further leaf of the Oommen-Rueda tree corresponding to a next character of plaintext;

(g) calculating a further Assignment Value for at least one further traversed branch of the Oommen-Rueda tree related to a further distribution of the further ciphertext previously calculated;

(h) calculating a further character of ciphertext relating to the further Assignment Value for the at least one further traversed branch of the Oommen-Rueda tree;

(i) repeating steps (e) through (h) until all of the plaintext has been processed; and

(j) outputting the ciphertext.

2. (withdrawn) A method for creating ciphertext from plaintext comprising the steps of:

(a) creating an Oommen-Rueda Tee;

(b) receiving a character of plaintext;

(c) traversing the Oommen-Rueda Tree between the root and that leaf corresponding to that character of plaintext and recording the Assignment Value of each branch so traversed;

(d) receiving a next character of plaintext; and

(e) repeating steps c and d until the plaintext has been processed.

3. (withdrawn) A method for creating ciphertext from plaintext comprising the steps of:

(a) receiving an Oommen-Rueda Tree;

(b) receiving a character of plaintext;

(c) traversing the Oommen-Rueda Tee between the root and that leaf corresponding to that character of plaintext and recording the Assignment Value of each branch so traversed;

(d) receiving a next character of plaintext; and

(e) repeating steps c and d until the plaintext has been processed.

4. (withdrawn) A method for creating ciphertext from plaintext comprising the steps of:

(a) creating an Oommen-Rueda Tree, which Oommen-Rueda Tree has leaves associated with the members of the alphabet of the plaintext, each member of the alphabet of the plaintext being associated with at least one leaf, which Oommen-Rueda Tree's internal nodes each have at least one branch depending therefrom, which Oommen-Rueda Tree branches have associated therewith an Assignment Value, which Assignment Value is associated with a member of the alphabet of the ciphertext, which Oommen-Rueda Tree's nodes each have associated therewith a quantity related to the frequency weight of each of the nodes and leaves dependant therefrom;

(b) receiving a character of plaintext;

(c) traversing the Oommen-Rueda Tee between the root and that leaf corresponding to that character of plaintext and recording the Assignment Value of each branch so traversed;

(d) receiving the next character of plaintext; and

(e) repeating steps c and d until the plaintext has been processed.

5. (withdrawn) A method for creating ciphertext from plaintext comprising the steps of:

(a) receiving an Oommen-Rueda Tee, which Oommen-Rueda Tree has leaves associated with the members of the alphabet of the plaintext, each member of the alphabet of the plaintext being associated with at least one leaf, which Oommen-Rueda Tree's internal nodes each have at least one branch depending therefrom, which Oommen-Rueda Tree branches have associated therewith an Assignment Value, which Assignment Value is associated with a member of the alphabet of the ciphertext, which Oommen-Rueda Tree's nodes each have associated therewith a quantity related to the frequency weight of each of the nodes and leaves dependant therefrom;

(b) receiving a character of plaintext;

(c) traversing the Oommen-Rueda Tree between the root and that leaf corresponding to that character of plaintext and recording the Assignment Value of each branch so traversed;

(d) receiving the next character of plaintext; and

(e) repeating steps c and d until the plaintext has been processed.

6. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4 or 5~~claim 1, wherein the Assignment Value for at least one branch traversed is determined in accordance with a Branch Assignment Rule.

7. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when a member of the ciphertext alphabet is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet to at least one of the branches being traversed between the root and the leaf so that that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

8. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when a member of the ciphertext alphabet is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet more frequently than other members of the ciphertext alphabet to the branches being traversed between the root and the leaf so that that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

9. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a zero to the majority of branches being traversed between the root and the leaf when zero is under-represented in the ciphertext generated thus far, and assigns

a one to the majority of branches being traversed between the root and the leaf when one is under-represented in the ciphertext generated thus far.

10. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein, when the conditional frequency of one member of the ciphertext alphabet given a particular sequence of members of the ciphertext alphabet in the ciphertext generated thus far, is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet to at least one of the branches being traversed between the root and the leaf so that the said conditional frequency of that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

11. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein, when the conditional frequency of one member of the ciphertext alphabet given a particular sequence of members of the ciphertext alphabet in the ciphertext generated thus far, is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet more frequently than other members of the ciphertext alphabet to the branches being traversed between the root and the leaf so that the said conditional frequency of that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

12. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein the Branch Assignment Rule assigns a member of the ciphertext alphabet to at least one of the branches being traversed between the root and the leaf, such assignment being determined by comparing a number associated with the frequency of at least one member of the ciphertext alphabet in the ciphertext generated thus far, with a number associated with the output of a pseudo-random number generator.

13. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when the ciphertext alphabet is binary, the

Branch Assignment Rule assigns a member of the binary alphabet to at least one of the branches being traversed between the root and the leaf, such assignment being determined by comparing a number associated with the frequency of a member of the binary alphabet in the ciphertext generated thus far, with a number associated with the output of a pseudo-random number generator.

14. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein the Branch Assignment Rule assigns a member of the ciphertext alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by a number associated with the output of a pseudo-random number generator.

15. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by comparing a number associated with the a pseudo-random number with a range equal to half the domain of the generator generating the pseudo-random number.

16. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein the Branch Assignment Rule assigns a member of the ciphertext alphabet of cardinality R to at least one branch being traversed between the root and the leaf, such assignment being determined by invoking at least two times (R minus 1) pseudo-random numbers, the domains of at least one of the pseudo-random numbers being related to the frequencies of the occurrences of the ciphertext characters generated thus far, and the domain of at least one of the other of the pseudo-random numbers having a mean of i/R for the ith branch of each node encountered in the traversal, where i is the relative position of the branch quantified by a pre-specified ordering of the branches, and the Branch Assignment Rule being such that where the ciphertext character associated with the ith branch in the said ordering is under-represented in the ciphertext generated thus far, it is no longer as under-represented.

17. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when the ciphertext alphabet is binary; the Branch Assignment Rule assigns a member of the binary ciphertext alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by invoking at least two pseudo-random numbers, the domain of the first of these pseudo-random numbers being related to the frequency of the occurrence of zero in the ciphertext, and the domain of a second of these pseudo-random numbers having a mean of 0.5, and the Branch Assignment Rule being such that when any character of the ciphertext alphabet is under-represented in the ciphertext generated thus far, it is no longer as under-represented.

18. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary ciphertext alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by comparing at least the output of two invoked pseudo-random numbers, the first of which has a domain having a mean between a number associated with the frequency of zeros and the quantity 0.5, and the second of which is a pseudo-random number having a domain whose mean is 0.5, and the Branch Assignment Rule being such that where any member of the ciphertext alphabet is under-represented in the binary ciphertext generated thus far, it is no longer as under-represented.

19. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 6 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary alphabet to at least one branch being traversed between the root and the leaf by utilizing at least two pseudo-random numbers, zero being assigned when a first pseudo-random number is less than a second pseudo-random number, where the generation of the second pseudo-random number is bounded between a number associated with the frequency of zeros in the ciphertext generated thus far and the

quantity of one minus the said number associated with the frequency of zeros in the ciphertext generated thus far.

20. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 or 19~~ claims 1 or 6 comprising the further steps of, after at least one traversal of the Oommen-Rueda Tree, recalculating a number associated with the frequency weight of at least one of the nodes of the Oommen-Rueda tree including the internal nodes and the leaves depending therefrom, and thereafter restructuring the Oommen-Rueda Tree in accordance with a Tree Restructuring Rule.

21. (previously presented) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claims 12, 13, 14, 15, 16, 17, 18, or 19, comprising the further step of receiving first key data associated with an initial seed for at least one of the generators of the pseudo-random numbers utilized by the Branch Assignment Rule.

22. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, comprising the further step of receiving second key data associated with the structure and labeling of the Oommen-Rueda Tree.

23. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, wherein the plaintext is modified prior to processing by the addition of a pre-specified prefix data stream.

24. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19;~~claims 1 or 6, wherein at least one of the steps is preformed by a suitably programmed processor.

25. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, wherein at least one of the steps is a process executed in software.

26. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, wherein at least one of the steps is a process executed in firmware.

27. (previously presented) A method for decoding ciphertext encoded utilizing an Oommen-Rueda Tree and utilizing a Branch Assignment Rule to provide Assignment Values for the branches depending from the root of the Oommen-Rueda Tree, comprising the steps of:

(a) receiving a first character of ciphertext;

(b) utilizing an Oommen-Rueda Tree having a structure corresponding to the Oommen-Rueda Tree initially utilized to encode the ciphertext and utilizing the same Branch Assignment Rule as utilized to encode the ciphertext to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed;

(c) if a leaf has not been reached, utilizing the same Branch Assignment Rule as utilized to encode the ciphertext to provide Assignment Values for the branches depending from the node that has been reached, receiving the next character of ciphertext, and continuing to traverse the Oommen-Rueda Tree from the node that has been reached towards a leaf, the current symbol of ciphertext determining the branch to then be traversed;

(d) when a leaf is reached, recording the plaintext character associated with the label of the leaf, the root becoming the node that has been reached for the purpose of further processing;

(e) repeating steps c and d until all symbols of ciphertext have been processed.

28. (previously presented) A method for decoding ciphertext, comprising the steps of:

(a) creating an Oommen-Rueda Tree structure corresponding to the Oommen-Rueda Tree initially utilized by the Encoder;

(b) receiving a first character of ciphertext;

(c) utilizing the Oommen-Rueda Tree structure, and utilizing the same Branch Assignment Rule as utilized by the Encoder to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed;

(d) if a leaf has not been reached, utilizing the same Branch Assignment Rule as utilized by the Encoder to provide Assignment Values for the branches depending from the node that has been reached, receiving the next character of ciphertext, and continuing to traverse the Oommen-Rueda Tree from the node that has been reached towards a leaf, the current symbol of ciphertext determining the branch to then be traversed;

(e) when a leaf is reached, recording the plaintext character associated with the label of the leaf, the root becoming the node that has been reached for the purpose of further processing;

(f) repeating steps d and e until all symbols of ciphertext have been processed.

29. (withdrawn) A method for decoding ciphertext, comprising the steps of:

(a) receiving data corresponding to the Oommen-Rueda Tree structure initially utilized by the Encoder, to create an Oommen-Rueda Tree having a structure corresponding- to the Oommen-Rueda Tee initially utilized by the Encoder;

(b) receiving a first character of ciphertext;

(c) utilizing the Oommen-Rueda Tree having a structure corresponding to the Oommen-Rueda Tree initially utilized by the Encoder and utilizing the same Branch Assignment Rule as utilized by the Encoder to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed;

(d) if a leaf has not been reached, utilizing the same Branch Assignment Rule as utilized by the Encoder to provide Assignment Values for the branches depending from the node that has been reached, receiving the next character of ciphertext, and continuing to traverse the Oommen-Rueda Tree from the node that has been reached towards a leaf, the current symbol of ciphertext determining the branch to then be traversed;

(e) when a leaf is reached, recording the plaintext character associated with the label of the leaf, the root becoming the node that has been reached for the purpose of further processing;

(f) repeating steps d and e until all symbols of ciphertext have been processed.

30. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28 or 29~~claims 27 or 28, wherein when the Encoder has utilized a Tree Restructuring Rule according to claim 20, after at least one traversal of the Oommen-Rueda Tree to a leaf, recalculating a number associated with the frequency weight of at least one of the nodes of the Oommen-Rueda Tree including the nodes and the leaves depending therefrom, the Oommen-Rueda Tree thereafter being structured in accordance with the same Tree Restructuring Rule utilized by the Encoder.

31. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, comprising the further step of receiving first key data associated with the initial seed for at least one of the generators of the pseudo-random numbers utilized by the Branch Assignment Rule, where first key data is associated with the first key data utilized by the Encoder.

32. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, comprising the further step of receiving second key data associated with the structure and labeling of the Oommen-Rueda Tree, where second key data is associated with the second key data utilized by the Encoder.

33. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, wherein when the plaintext has been modified prior to processing by the addition of a pre-specified prefix data stream, the ciphertext is modified prior to processing by the addition of an encoded pre-specified data stream in a prefix manner, and where this encoded pre-specified data stream is related to the pre-specified data stream utilized by the Encoder.

34. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, wherein at least one of the steps is preformed by a suitably programmed processor.

35. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, wherein at least one of the steps is a process executed in software.

36. (Currently Amended) A method for decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, wherein at least one of the steps is a process executed in firmware.

37. (withdrawn) A device for creating ciphertext from plaintext comprising:

(a) receiving means for receiving a plaintext data stream;

(b) storing means for storing data representative of an Oommen-Rueda Tree;

(c) processing means for processing the plaintext data stream on a character by character basis, and for repeatedly traversing the stored Oommen-Rueda Tree for each character of plaintext to be processed, each such traversal being between the root and that leaf corresponding to that symbol of plaintext then being processed;

(d) recording means for recording the Assignment Values of the branches of the Oommen-Rueda Tree traversed during the processing of the plaintext data stream; the receiving means, the storing means and the recording means being in communication with the processing means.

38. (withdrawn) A device for creating ciphertext from plaintext according to claim 37, further comprising a second storing means in communication with the processing means for storing a Branch Assignment Rule, and wherein the processing means additionally calculates a number associated with the frequency of at least one member of the ciphertext alphabet in the ciphertext generated thus far, and calculates the Assignment Value of at least one branch depending from at least one node reached in processing.

39. (withdrawn) A device for decoding ciphertext comprising:

(a) receiving means for receiving a ciphertext data stream;

(b) storing means for storing data representative of the Oommen-Rueda Tree utilized by the Encoder;

(c) processing means for:

1. processing the ciphertext data stream on a character by character basis,

2. repeatedly traversing the stored Oommen-Rueda Tree between the root and leaves of the Oommen-Rueda Tree, the choice of branch to traverse being determined by the comparison of the character of ciphertext then being processed and the Assignment Value of such branch;

3. selecting the next character of ciphertext for processing to determine the next branch to be traversed, until a leaf is reached, whereupon the plaintext character associated with the leaf is recorded, and the processing continues from the root; and

4. repeating the above steps (1), (2) and (3) until all ciphertext to be processed has been processed; (d) recording means for recording the plaintext associated with each of the leaves reached during processing; the receiving means, the storing means and the recording means being in communication with the processing means.

40. (withdrawn) A device for decoding ciphertext according to claim 39 further comprising a second storing means in communication with the processing means, for storing a Branch Assignment Rule, and wherein the processing means additionally calculates a number associated with the frequency of at least one member of the ciphertext alphabet in the ciphertext processed thus far, and calculates the Assignment Value of at least one branch depending from at least one node reached in processing.

41. (Currently Amended) A secure communication method comprising the steps of:

(a) encoding the communication in accordance with a method described in ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, herein;

(b) transmitting the encoded communication;

(c) receiving the encoded communication; and

(d) decoding the encoded communication.

42. (Currently Amended) A secure data storage and retrieval method comprising the steps of:

(a) encoding the data in accordance with a method described in ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, herein;

(b) storing the encoded data;

(c) retrieving the encoded data; and

(d) decoding the encoded data.

43. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, further comprising the step of subsequently encrypting the ciphertext generated by any standard encryption process.

44. (Currently Amended) A method of decoding ciphertext according to ~~claims 27, 28, or 29~~claims 27 or 28, comprising the further steps of initially decrypting the ciphertext data stream using the standard decryption process associated with the encryption process, and thereafter decoding.

45. (Currently Amended) A method for embedding a plaintext message into carrier data comprising the steps of converting the plaintext into ciphertext in accordance with ~~claims 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, or 19~~claims 1 or 6, and thereafter steganographically inserting the ciphertext into the carrier data.

46. (previously presented) A method for extracting a plaintext message from carrier data which carrier data contains embedded ciphertext steganographically inserted into carrier data in accordance with claim 45, comprising the steps of steganographically extracting the ciphertext from the carrier data, and further decoding the ciphertext utilizing a process in accordance with claims 27, 28, or 29.

47. (original) A method for generating pseudo-random numbers comprising the steps of receiving an arbitrary data stream, and processing the data stream in accordance with the process of claims 12, 13, 14, 15, 16, 17, 18, or 19.

48. (previously presented) The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 1, wherein the ciphertext possesses Statistical Perfect Secrecy.

49. (previously presented) The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 1, wherein the ciphertext output is at least one of displayed, transmitted, and stored.